Vinum Analytica: Wine Review Insight

Paolo Palumbo

September 12, 2024

Abstract

This report presents a data mining project that analyzes the language used in wine reviews written by sommeliers. We compare the performance of three different models: Decision Tree, Random Forest, and Neural Network. The analysis includes a detailed examination of preprocessing techniques, class contamination removal, and feature vectorization using TF-IDF. We assess the models based on their accuracy and present the results through various evaluations.

1 Introduction

The growing availability of online wine reviews offers a valuable resource for understanding the language used to describe different wine varieties. This study aims to analyze the vocabulary employed in these reviews to classify wines based on the grape variety being reviewed. By examining the textual features of the reviews, we seek to uncover linguistic patterns that correlate with specific grape varieties. Three machine learning models—Decision Tree, Random Forest, and Neural Network—are utilized to explore and classify the relationship between the language used and the wine variety.

2 Corpus Description

The dataset used in this study consists of 130,000 wine reviews, sourced from the Wine Enthusiast website via Kaggle. Each review includes detailed information about the wines, such as the variety, origin, price, and the reviewer's evaluation. The primary focus of this dataset is the textual description of the wines, where reviewers provide sensory insights into the wines' characteristics, such as aroma, flavor, and body.

Not all reviews in the dataset contain complete information, and for the purposes of this study, reviews missing key data, such as the grape variety, were excluded. This cleaned dataset provides a rich source of textual data for analyzing how different grape varieties are described and for training models to classify wines based on these descriptions.

2.1 Exploratory Data Analysis

The dataset contains a diverse range of wine varieties, with some being more common than others.

To ensure a clearer and more coherent representation of wine varieties, a selection was made based on the presence of well-defined grape varieties. Specifically, varieties labeled as 'Blend' were excluded. These varieties represent wines made from blends of different grape varieties rather than a single variety. Their exclusion was motivated by the desire to analyze specific and well-defined grape varieties, avoiding the ambiguity introduced by blends.

Additionally, a minimum representation threshold was applied to the remaining varieties, ensuring that only those with an adequate number of reviews were included in the analysis. This approach allowed for a dataset focused on clearly identified and sufficiently represented wine varieties, facilitating a more accurate and meaningful analysis. The selection helped avoid distortions due to inadequate representation and ensured that the results obtained are based on robust and representative data.

As a result of this selection process, the dataset now includes **29** distinct grape varieties, namely *Portuguese Red*, *Pinot Gris*, *Riesling*, *Pinot Noir*, *Gewürztraminer*, *Cabernet Sauvignon*, *Chardonnay*, *Malbec*, *Merlot*, *Gamay*, *Sauvignon Blanc*, *Sangiovese*, *Cabernet Franc*, *Petite Sirah*, *Rosé*, *Zinfandel*, *Grüner Veltliner*, *Viognier*, *Syrah*, *Nebbiolo*, *Barbera*, *Portuguese White*, *Sangiovese Grosso*, *Shiraz*, *Grenache*, *Pinot Grigio*, *Tempranillo*, *Glera*, and *Port*. After filtering, the cleaned dataset comprises **85,117** reviews, providing a well-defined and comprehensive basis for further analysis.

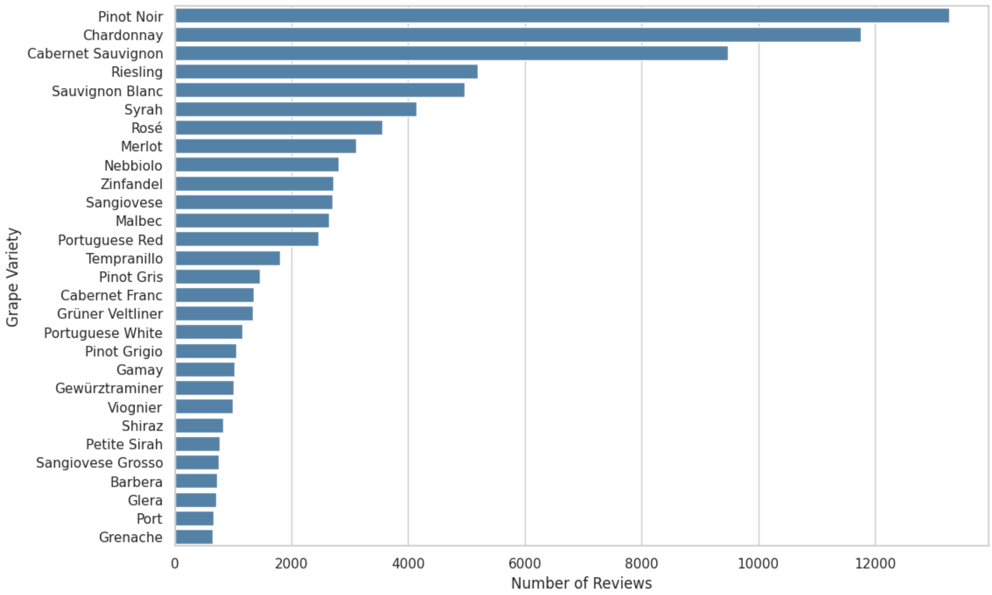

Figure 1: Most represented grapes varieties in the dataset

2.2 Contamination Removal

To further refine the dataset, the keyword for each grape variety was removed from the descriptions. This was done to eliminate any bias introduced by the presence of the variety name in the text. By removing the variety keyword, the analysis can focus solely on the sensory characteristics and descriptions of the wines, without being influenced by the specific grape variety being reviewed.

3 Preprocessing

Data preprocessing is a critical step in any machine learning application. In this project, we applied several preprocessing techniques to prepare the text data for modeling:

- **Text Normalization:** All text was converted to lowercase to ensure uniformity. Additionally, Unicode characters were transformed into ASCII to handle special characters and maintain consistency across the dataset.
- **Tokenization:** Text was split into individual words or tokens.
- **Stop Words Removal:** Common words that do not contribute to the semantic meaning were removed.
- **Stemming/Lemmatization:** Words were reduced to their base or root form.

As shown in Figure 1, the dataset is imbalanced. The most common grape variety is Pinot Noir, and the least common is Grenache.

The dataset was split into a training set and a test set with a 80% - 20% ratio.

3.1 Feature Vectorization

To transform the text data into a numerical format suitable for machine learning, we used the TF-IDF (Term Frequency-Inverse Document Frequency) method. This approach captures the importance of words in the context of the entire dataset and helps in feature extraction.

4 Modeling and Hyperparameter Tuning

Three different categories of models were considered:

- *Decision Trees*
- *Random Forests*
- *Neural Networks*

4.1 Imbalanced Dataset

To overcome the class imbalance in the dataset, the training set was oversampled using the SMOTE technique. The number of records was limited to 200,000 to control computational time and resources.

SMOTE generates synthetic samples for the minority classes by interpolating between existing samples, which helps to balance the class distribution without duplicating data. This approach improves the model's ability to generalize, especially for underrepresented classes, while reducing the risk of overfitting. In addition to SMOTE, class weights were assigned to further address the imbalance during training, ensuring that minority classes had adequate influence on the model's learning process.

4.2 Model Selection

To identify the optimal hyperparameter settings for each model, we performed a random search by exploring 8 randomly selected hyperparameter combinations for each model (Decision Trees, Random Forests, and Neural Networks). Each combination was evaluated using 6-fold cross-validation on the training set. The hyperparameter search spaces for the first phase are outlined in Table 1, Table 2, and Table 3.

Accuracy was used as the metric to evaluate the models.

Table 1: Hyperparameter Grid for Decision Tree

Hyperparameter	**Values**
Criterion	gini, entropy, log_loss
Min Impurity Decrease	0.0, 1e-4, 1e-8, 1e-12
Max Depth	None, 1000

Table 2: Hyperparameter Grid for Random Forest

Hyperparameter	**Values**
Number of Estimators	50, 100, 150
Criterion	gini, entropy, log_loss
Min Impurity Decrease	0.0, 1e-4, 1e-8, 1e-12
Max Depth	None, 1000

Table 3: Hyperparameter Grid for Neural Network

Hyperparameter	**Values**
Hidden Size	16, 32, 64
Epochs	5, 10, 15
Learning Rate	0.005, 0.001, 0.01

5 Experimental Results and Analysis

5.1 Hyperparameter Selection

For each model, we selected the hyperparameter set that achieved the highest average accuracy over the 6-fold cross-validation.

Once the best hyperparameters for each model were identified, we employed the Wilcoxon test to determine whether the differences between the average accuracies of the models were statistically significant. This step ensured that the observed differences were not due to random variation.

Finally, we compared the best results obtained from the three models (Decision Tree, Random Forest, and Neural Network) to assess which model exhibited the highest overall performance.

5.1.1 Decision Tree Results

The Decision Tree model was evaluated with various hyperparameters. Among the configurations tested, the best performance was achieved with a maximum depth of 1000, using the ”gini” criterion and a minimum impurity decrease of 0.0. This setting resulted in the highest average accuracy of approximately 0.4578. The models using the ”entropy” criterion and ”log_loss” criterion with a maximum depth of 1000 also performed well, with accuracies around 0.4199 and 0.4136, respectively. On the other hand, configurations with a maximum depth of *null* and ”entropy” or ”log_loss” criteria yielded lower accuracies, indicating that deeper trees tend to generalize better in this scenario.

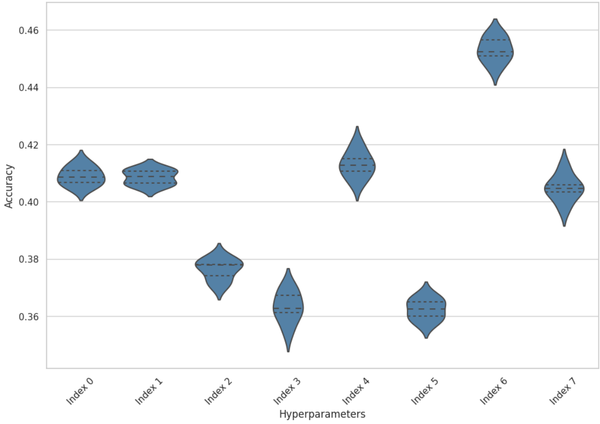

Figure 2: Accuracy distribution for Decision Trees

5.1.2 Random Forest Results

The Random Forest model demonstrated its best performance with a configuration of 100 estimators, "gini" criterion, and a maximum depth of *null*. This setup achieved the highest average accuracy of approximately 0.6267. Other effective configurations included 100 estimators with the "gini" criterion and a maximum depth of 1000, reaching an accuracy of around 0.6241. The "log_loss" criterion with 150 estimators also performed well but was slightly less accurate compared to the "gini" based configurations. Conversely, models with fewer estimators or different criteria, such as "entropy," generally resulted in lower accuracies. This indicates that increasing the number of estimators and optimizing depth are crucial for improving performance in Random Forest models.

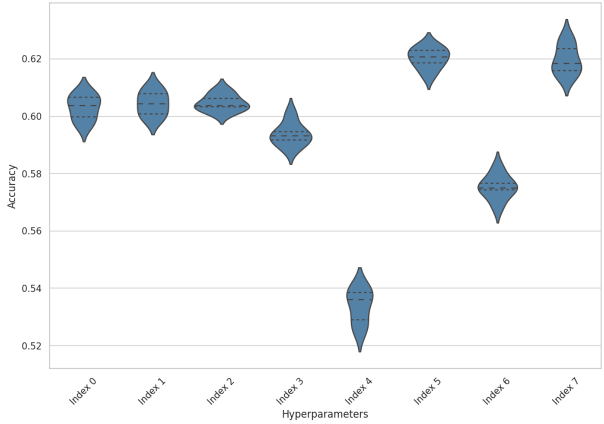

Figure 3: Accuracy distribution for Random Forests

5.1.3 Neural Network Results

For the Neural Network models, the configurations with a hidden size of 64 and learning rates of 0.005 or 0.01 yielded the highest average accuracies. Specifically, the setting with 64 hidden units, 10 epochs, and a learning rate of 0.005 achieved an accuracy of around 0.6097, while another setting with the same hidden size and learning rate of 0.01 resulted in an accuracy of approximately 0.6136. The best performance was observed with a configuration of 64 hidden units, 5 epochs, and a learning rate of 0.001, achieving an accuracy of 0.6311. In contrast, models with smaller hidden sizes and shorter epochs generally showed lower performance, suggesting that deeper networks with more training epochs are more effective in capturing complex patterns.

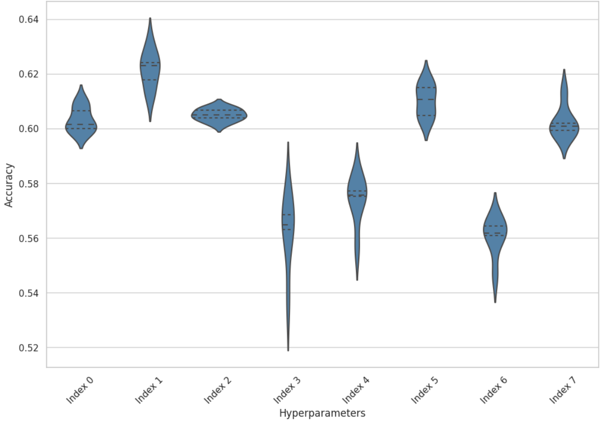

Figure 4: Accuracy distribution for Neural Network

5.2 Best Model Comparison

Among the evaluated models, the Neural Network (NN) configuration with 64 hidden units, 5 epochs, and a learning rate of 0.001 achieved the highest average accuracy of approximately 0.6244. This model demonstrates strong performance, particularly in its ability to capture complex patterns within the data. The Random Forest (RF) model, with 100 estimators, "gini" criterion, and a maximum depth of 1000, also performed impressively, attaining an average accuracy of around 0.6219. Although slightly lower than the NN, it remains a robust model for classification tasks. The Decision Tree (DT) model, using the "gini" criterion, a minimum impurity decrease of 1e-08, and a maximum depth of 1000, had an average accuracy of about 0.4579. While this accuracy is significantly lower than that of the RF and NN models, it is important to note that Decision Trees are generally less effective on their own compared to ensemble methods and neural networks. Overall, the Neural Network achieved the highest performance, followed closely by the Random Forest, while the Decision Tree showed comparatively lower results.

6 Future Work

6.1 Enhancing the Current Approach

Although the current models offer promising results, several improvements could be made. Fine-tuning the hyperparameters of the Neural Network model is one potential avenue for enhancing its performance. Techniques such as learning rate scheduling, advanced regularization methods, and experimenting with different network architectures could lead to better generalization and reduced overfitting. Ad-

Index	**Criterion**	**Min Impurity Decrease**	**Max Depth**
0	log_loss	0.0	1000
1	entropy	1e-12	1000
2	gini	0.0001	1000
3	entropy	0.0001	None
4	entropy	1e-12	None
5	log_loss	0.0001	None
6	gini	1e-08	1000
7	entropy	1e-08	1000

Table 4: Hyperparameters for Decision Tree

Index	**n_estimators**	**criterion**	**max_depth**
0	150	log_loss	None
1	50	gini	None
2	50	gini	1000
3	100	entropy	None
4	50	entropy	None
5	100	gini	1000
6	50	log_loss	None
7	100	gini	None

Table 5: Hyperparameters for Random Forest

Index	**Hidden Size**	**Epochs**	**Learning Rate**
0	64	10	0.005
1	64	5	0.001
2	64	15	0.005
3	16	5	0.01
4	16	5	0.005
5	64	5	0.005
6	16	15	0.01
7	64	10	0.01

Table 6: Hyperparameters for Neural Network

ditionally, exploring more sophisticated methods for handling class imbalance, such as ADASYN or incorporating class-specific loss functions, could further improve model accuracy, particularly for underrepresented classes.

6.2 Regression Analysis on Wine Prices

In addition to improving the existing classification models, future work could extend the analysis to a regression task focused on predicting wine prices. Implementing regression models to predict continuous variables, such as wine prices, would involve developing and training models specifically tailored for regression tasks, such as Linear Regression, Decision Trees for Regression, Random Forest Regressors, and Neural Networks with regression output layers.

The inclusion of price prediction could offer valuable insights into the factors that influence wine pricing, beyond just classification of wine varieties. This would involve preprocessing data for regression, exploring feature engineering techniques, and evaluating model performance using appropriate regression metrics (e.g., Mean Absolute Error, Mean Squared Error). Combining these regression results with the current classification models could provide a more comprehensive understanding of wine reviews and their impact on pricing.

7 Conclusion

In this study, we investigated various machine learning models for analyzing wine reviews, focusing on classification tasks related to wine varieties. We compared the performance of Decision Trees, Random Forests, and Neural Networks using a rigorous hyperparameter tuning process and cross-validation. Our results indicated that while Decision Trees provided a solid baseline, Random Forests offered improved accuracy and robustness by leveraging ensemble methods. The Neural Networks demonstrated the highest potential for capturing complex relationships within the data, albeit requiring more computational resources and careful tuning to prevent overfitting.

Looking ahead, there are several avenues for further research. Enhancements to the current models, such as advanced hyperparameter tuning and alternative techniques for handling class imbalance, could yield even better results. Additionally, extending the analysis to a regression task to predict wine prices could provide valuable insights into the factors influencing wine pricing. This future work would involve developing regression models, refining data preprocessing, and evaluating performance with regression-specific metrics.

Overall, the findings of this study contribute to a deeper understanding of wine reviews and set the stage for future improvements and expansions in this domain.